Enhancing Geographic Greedy Routing in Sparse LDACS Air-to-Air Networks through k-Hop Neighborhood Exploitation

Abstract—**The emergence of the L-band Digital Aeronautical Communications System (LDACS) presents a significant opportunity for enabling Air-to-Air (A2A) communication to accommodate the growing number of aircraft. However, deploying LDACS entails overcoming challenges, including mitigating large Medium Access Control (MAC) delays and improving multi-hop connectivity across sparse networks. Geographic greedy routing, commonly used in Aeronautical Ad-hoc networks, utilizes position information to eliminate the need for topology discovery. But, as network density decreases, greedy forwarding increasingly fails. With the gradual introduction of aircraft equipped with LDACS, it becomes crucial to improve greedy forwarding performance. This research investigates the application of greedy forwarding with k-hop neighborhood information, denoted by Greedy-k, to enhance performance specifically in sparse networks. Our study presents a novel approach that reduces beacon size by transmitting only a subset of the k-hop neighborhood, ensuring the information fits within an LDACS time slot. We derived the subset size analytically and evaluated the performance through simulations benchmarked against the conventional Greedy-1 method with respect to the French airspace scenario. Our results indicate that the proposed approach achieves up to 13% higher Packet Delivery Ratio (PDR) than Greedy-1, while capturing additionally 70.1% and 34.6% of 2$^{\text{nd}}$ and 3$^{\text{rd}}$ order neighbors, respectively.**

Index Terms—**geographic routing, greedy forwarding, A2A, LDACS**

I. Introduction

By the year 2050, the European Organisation for the Safety of Air Navigation (EUROCONTROL) forecasts a 44% increase in flight numbers compared to 2019, highlighting the urgent need for advanced communication technologies that meet rising demands for safety, data rate, and reliability [1]. The L-band Digital Aeronautical Communications System (LDACS), currently under standardization, is proposed as a solution for Air-to-Ground (A2G) communication within the Future Communications Infrastructure (FCI), potentially enhancing data rates by up to 200% [2]. Adapting LDACS for Air-to-Air (A2A) communication within the FCI remains a key area of research [3].

Recently, the Multi-Channel Self-Organized Time-Division Multiple Access (MCSOTDMA) protocol has been proposed as a viable Medium Access Control (MAC) layer for LDACS A2A communications [4]. This protocol utilizes a Shared (SH) channel for broadcasting beacons and multiple Point-to-Point (PP) channels for unicast packets, ensuring deterministic delays. The state-of-the-art routing protocol for A2A communication is geographic routing [5]. It utilizes beacons to broadcast positional information and greedily routes packets towards their destination. When a packet reaches a dead-end, it is either dropped or requires a backup mechanism to proceed. In deploying LDACS for A2A communications, we identify two key challenges.

Challenge I: The practical implementation of LDACS for A2A links is expected to be a long process, starting by integrating the technology into newly built aircraft. The airspace will include both legacy and LDACS-equipped aircraft until full LDACS deployment is achieved. This situation raises concerns about LDACS A2A performance in sparse networks, particularly regarding the coexistence with legacy aircraft over the coming decades.

Challenge II: In the SH channel, a Randomized Slotted ALOHA (RS-ALOHA) is used. This results in a MAC delay that increases linearly with the number of neighboring aircraft [6]. Using a high communication range increases the number of direct neighbors, resulting in significant MAC delays affecting both SH and PP links. To accommodate future growth in airspace, reducing the communication range while ensuring high connectivity with ground stations can mitigate these delays. However, this reduction requires improvements in geographic greedy routing to ensure routes are found when they exist topologically.

This paper proposes a solution that can cope with both difficulties. Greedy forwarding, a main component of geographic routing, proves to be highly efficient in high-density airspace scenarios, achieving performance comparable to that of the shortest path routing [7]. As network density decreases, conventional greedy forwarding increasingly faces difficulties, due to frequent dead-ends, as will be discussed in Figure 6. To improve performance in sparse scenarios, Greedy-k forwarding, which utilizes knowledge of the k-hop neighborhood, is proposed to enhance the efficiency of Greedy-1 [8], [9]. However, this approach increases beacon sizes beyond the capacity of an LDACS A2A slot in the SH channel. Therefore, we introduce a method that incorporates a subset of 1-hop neighborhood within beacon messages, providing partial knowledge of the 2-hop neighborhood. Furthermore, our approach leverages unused slots in the beacon to share additional neighbors at exactly 2-hop distance, which extends our knowledge to a subset of 3-hop neighborhood. In the remainder of this paper, we refer to neighbors at exactly k-hop distance as k^{th} order neighbors.

Our main contributions are:

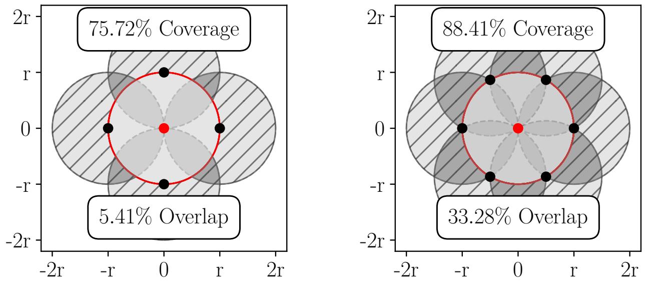
- We analytically derive an optimal fixed subset size required to achieve performance improvements with Greedy-k forwarding.
- We propose three methods (*Random*, *Farthest First (FF)* traversal and *Extended Farthest First (EFF)* traversal) for selecting subsets of neighbors at exactly k-hop distances and introduced a geographic greedy routing protocol that leverages this neighborhood information.
- We evaluate the performance of our proposed routing protocol and the selection methods over the French airspace with different equipage fractions. This reflects the initial use of LDACS equipped aircraft within the next few years and its extended utilization projected for the coming decades.

The organization of the paper is as follows: The required subset size is discussed in Section II. The proposed geographic routing protocol is introduced in Section III. The simulation setup is outlined in Section IV. Our results and discussions are detailed in Section V. Related work is discussed in Section VI. The conclusion and further insights on future directions are given in Section VII.

II. The Subset Size of the k-Hop Neighborhood

To achieve performance gains in Packet Delivery Ratio (PDR) with Greedy-k routing, a node requires only a subset of its k-hop neighborhood. However, the challenge is determining how large this subset should be. Consider a network denoted as $\mathcal{A}$, consisting of nodes referred to as a_i, for all $a_i \in \mathcal{A}$. Let $\mathcal{N}^1_{a_i}$ represent the set of 1st order neighbors of a_i. The cardinality denoted by $|\cdot|$ represents the number of elements within a given set. Define $\mathcal{S}^1_{a_i}$ as a subset of $\mathcal{N}^1_{a_i}$, where the size of this subset is denoted by $m = |\mathcal{S}^1_{a_i}|$. Our goal is to identify the minimum required m that enables Greedy-k routing, while ensuring that the selected nodes are avoiding redundancy.

Coverage is modeled using a Unit Disc Graph (UDG) in Two-dimensional (2D) space, where two nodes are connected if within range r. Although aircraft move in Three-dimensional (3D) space, the altitude variation is minor compared to their communication range. Consider $\mathcal{N}^1_{a_i}$ as an infinite set, uniformly distributed along the circumference of the UDG, positioned to maximize their distance from a_i. Let the subset size $m = |\mathcal{S}^1_{a_i}| \in [0, \infty)$. When selecting m nodes from $\mathcal{S}^1_{a_i}$, we ensure these nodes are as distant from each other as possible to maximize diversity. For each node a_j in this diverse subset, we calculate the percentage of the coverage area beyond a_i's own, relative to the total area of 2nd order neighbors, i.e., $3\pi r^2$, denoted as *Coverage*. Additionally, we assess the extent of overlap within this additional area, denoted as *Overlap*. The case for $m = 4, 6$ is demonstrated in Figure 1, where the Coverage is depicted by the grey hashed area (75.72%, 88.41%) and Overlap by the dark grey hashed area (5.41%, 33.28%) respectively.



(a) $m = 4$.

(b) $m = 6$.

Fig. 1: Comparison of coverage and overlap areas for different subset lengths $m = |\mathcal{S}^1_{a_i}|$ where in red is the node a_i and r represent the communication range.

Furthermore, we analyzed various m values to assess their effects on coverage, overlap, the rate of change of coverage (ΔCoverage) and the rate of change of overlap (ΔOverlap), as depicted in Figure 2. Here, increasing m enhances coverage, yet ΔCoverage decreases after $m = 3$ as overlap among nodes starts. When $m > 4$, we see that ΔCoverage $<$ ΔOverlap, which means the redundancy

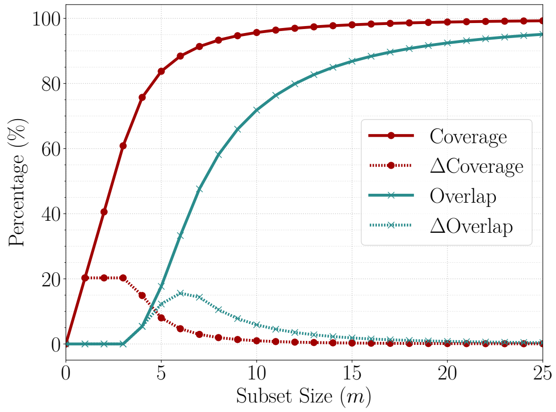


Fig. 2: The coverage and overlap area for different values for the subset of neighborhood length m and the rate of change of the coverage area ΔCoverage and overlap area ΔOverlap.

among selected nodes increases faster than the increase in coverage. In this study, we establish $m = |\mathcal{S}^1_{a_i}| = 4$ to optimally distribute coverage around a_i, maintaining ΔCoverage > ΔOverlap.

III. Geographic Greedy Routing with a Subset of k-Hop Neighborhood

A. System Model

Within the routing algorithm, a_c denotes the currently processed node. Each node $a_c \in \mathcal{A}$ receives beacons B_{a_s} from beacon senders $a_s \in \mathcal{N}^1_{a_c}$. Beacons include the sender's address a_s, position p_{a_s}, sequence number seq_{a_s}, and neighbor subset information. For each neighbor a_i in this subset, its address, position p_{a_i}, hop count h_{a_i}, and sequence number seq_{a_i} are in B_{a_s}. A beacon entry occupies 12 B, 4 B for the node address (1 B prefix plus 24 bit International Civil Aviation Organization (ICAO) ID), 6 B for the position in Compact Position Reporting (CPR) format, 1 B for hop count, and 1 B for the sequence number. Sequence numbers, incremented by each node $a_c \in \mathcal{A}$ upon $\mathcal{N}^1_{a_c}$ changes, track direct neighbor additions or removals.

The set $\mathcal{N}_{a_c}$ contains all neighbors in the position table of a node $a_c \in \mathcal{A}$. Moreover, denote the set of k^{th} order neighbors of a_c by $\mathcal{N}^k_{a_c}$. We also introduce $\mathcal{S}^k_{a_c} \subseteq \mathcal{N}^k_{a_c}$, a subset of the k^{th} order neighbors of a_c. In the position table $\mathcal{T}_{a_c}$, an entry for $a_i \in \mathcal{N}_{a_c}$ includes a_i's address, position p_{a_i}, hops to a_i h_{a_i}, sequence number seq_{a_i}, via address via_{a_i}, and validity time t_{a_i}. Entries are updated from beacons, with the via address indicating the sender of the beacon. Let $t_{B_{a_s}}$ represent the beacon reception time from a_s and $t_{validity}$ the validity interval, a protocol parameter. The validity time of a_i's entry, which is known via a_s, in $\mathcal{T}_{a_c}$ is given by,

$$t_{a_i} = t_{B_{a_s}} + t_{validity}. \tag{1}$$

We show in Table I an example of a position table belonging to a node a_c.

TABLE I: Position Table of a_c ($\mathcal{T}_{a_c}$)

$\mathbf{a}_i$	$\mathbf{p_{a_i}}$	$\mathbf{h_{a_i}}$	$\text{seq}_{\mathbf{a_i}}$	$\text{via}_{\mathbf{a_i}}$	$\mathbf{t_{a_i}}$
a_1	(x_1, y_1, z_1)	2	22	a_2	20.5 s
a_2	(x_2, y_2, z_2)	1	34	a_2	24 s
...	...	...	...	...	...

B. Neighbor Selection Mechanism

In this section, we describe three methods for selecting a subset $\mathcal{S}^k_{a_c} \subseteq \mathcal{N}^k_{a_c}$. These methods identify specific neighbors of a_c within exactly k-hop distance. The first method involves selecting a subset of size m through random selection, denoted as *Random*. While simple, this method serves as an initial baseline for comparison with more complex strategies. However, this approach does not consider the spatial location of the selected nodes. This oversight can lead to poor selection quality, especially when the number of elements in $\mathcal{N}^k_{a_c}$ is large.

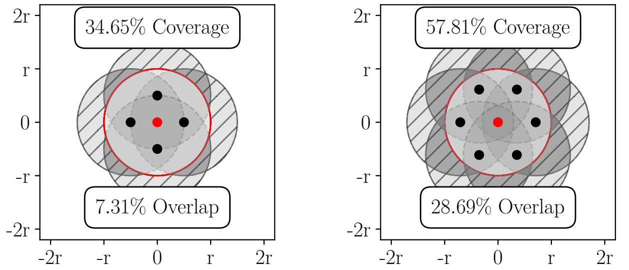
The second method we explore is the *FF* algorithm. The process starts with randomly selecting an initial node, then iteratively adds the node farthest from those already in the subset. The selection continues until the subset size reaches m or all nodes have been selected, ensuring optimal spatial diversity among the chosen nodes. The specifics of this method, where we use a threshold distance $d_{\text{thr}} = 0$, are outlined in Algorithm 1.

Algorithm 1: $\text{EFF}(\mathcal{N}^k_{a_c}, m, d_{\text{thr}}, \mathcal{S}^{\text{pre}}_{a_c})$

```
Input: Set of neighbors $\mathcal{N}^k_{a_c}$ of the current node
       $a_c$, subset size $m$, distance threshold $d_{\text{thr}}$,
       preselected set of nodes $\mathcal{S}^{\text{pre}}_{a_c} = \emptyset$ (default)
Output: Selected set of nodes $\mathcal{S}^k_{a_c}$
$d_{\text{thr,c}} \leftarrow \frac{1}{\sqrt{2}} \cdot d_{\text{thr}}$ // minimum distance to $a_c$
foreach $a_i$ in $\mathcal{N}^k_{a_c} \cup \mathcal{S}^{pre}_{a_c}$ do
    extract the position $p_{a_i}$ from $\mathcal{T}_{a_j}$
if $\mathcal{S}^{pre}_{a_c}$ is empty then
    Select arbitrarily $a_r$ in $\mathcal{N}^k_{a_c}$ and add it to $\mathcal{S}^k_{a_c}$
else
    $\mathcal{S}^k_{a_c} \leftarrow \mathcal{S}^{\text{pre}}_{a_c}$
// ensure $|\mathcal{S}^k_{a_c}| \leq m$
$m \leftarrow \min\{m, |\mathcal{N}^k_{a_c}| + |\mathcal{S}^{\text{pre}}_{a_c}|\}$ while $|\mathcal{S}^k_{a_c}| < m$ do
    $d_{\text{max}} \leftarrow$ -1
    foreach $a_i$ in $\mathcal{N}^k_{a_c} \setminus \mathcal{S}^k_{a_c}$ do
        $d_{a_i} \leftarrow \min\{\|p_{a_i} - p_{a_j}\|_2 \mid a_j \in \mathcal{S}^k_{a_c}\}$
        $d_{a_i,c} \leftarrow \|p_{a_i} - p_{a_c}\|_2$
        if $(d_{a_i} > d_{max})$ and $(d_{a_i} > d_{thr})$ and
         $(d_{a_i,c} > d_{thr,c})$ then
            // $d_{\text{thr}} = 0$, defaults to FF.
            $d_{\text{max}} \leftarrow d_{a_i}$ and $a_{\text{max}} \leftarrow a_i$
    if $d_{max}$ = -1 then
        break
    Add $a_{\text{max}}$ to $\mathcal{S}^k_{a_c}$
$\mathcal{S}^k_{a_c} \leftarrow \mathcal{S}^k_{a_c} \setminus \mathcal{S}^{\text{pre}}_{a_c}$
return $\mathcal{S}^k_{a_c}$
```

The *FF* algorithm can also incorporate a set of preselected elements $\mathcal{S}^{\text{pre}}_{a_c}$ as input. To select m elements, we invoke the algorithm with $\text{EFF}(\mathcal{N}^k_{a_c}, m, d_{\text{thr}} = 0, \mathcal{S}^{\text{pre}}_{a_c} = \emptyset)$, where an empty $\mathcal{S}^{\text{pre}}_{a_c}$ implies that selection starts with a randomly chosen node instead of from the preselected subset.

Both the *Random* and *FF* methods select up to m nodes, even when some of them are redundant. The last method, *EFF*, refines the *FF* by setting a threshold distance $d_{\text{thr}} = \frac{1}{\sqrt{2}} \cdot r$, adapted for UDG scaled to the communication range r. The choice of $\frac{1}{\sqrt{2}}$ ensures that the farthest nodes in the subset do not fall out of each other's communication range when four nodes are considered. Additionally, each node selected after the first must be at a minimum distance from the current node a_c of $d_{\text{thr},c} = \frac{1}{\sqrt{2}} \cdot d_{\text{thr}}$. This distance corresponds to half the diagonal of a square with side length d_{thr}, centered at a_c. This strategy prevents redundancy by avoiding the selection of nodes that are too close to each other, as depicted in Figure 3a for the case of $m = 4$ and Figure 3b for a subset size $m = 6$. For example, consider the case of selecting m nodes from $\mathcal{N}^1_{a_c}$ using *EFF*. Applying the thresholds d_{thr} and $d_{\text{thr},c}$ may result in selecting fewer than m nodes due to the spatial arrangement of these neighbors. This approach allows us to extend our selection to 2nd order neighbors $\mathcal{N}^2_{a_c}$, as will be further discussed in Section III-C. The entire process is detailed in Algorithm 1.



(a) $m = 4$, with $d_{\text{thr},c} = \frac{r}{2}$.

(b) $m = 6$, with $d_{\text{thr},c} = \frac{r}{\sqrt{2}}$.

Fig. 3: Coverage and overlap areas for subsets $\mathcal{S}^1_{a_c}$: nodes are distanced from the center a_c (displayed in red) by at least 0.5 and maintain a separation of $\frac{r}{\sqrt{2}}$ from each other.

C. Beaconing Mechanism

The proposed routing protocol operates by utilizing beacons to disseminate information regarding a subset of the local k-hop neighborhood of a node, as detailed in Section III-B. Initially, each beacon contains only information about the node itself. Over time, it accumulates and broadcasts additional details about its neighborhood. The received beacons are then added to the position table, as illustrated in Table I.

Each node broadcasts a new beacon at fixed intervals determined by the beacon interval parameter. The process initiates with the inclusion of the beacon sender's address a_s, position p_{a_s}, and sequence number seq_{a_s}. It is assumed that each node is aware of its own position. The sequence number helps mitigate routing loops among nodes within the k-hop neighborhood when a node is removed from the position table. When needing to send a beacon, a_s extracts two distinct sets, $\mathcal{N}^1_{a_s}$ and $\mathcal{N}^2_{a_s}$. Using the *Random* or *FF* methods, a node selects up to m neighbors directly from $\mathcal{N}^1_{a_s}$. With *EFF*, a subset of $\mathcal{N}^1_{a_s}$ is chosen using the defined d_{thr}. If fewer than m neighbors are selected from $\mathcal{N}^1_{a_s}$ using *EFF*, the remainder is selected from $\mathcal{N}^2_{a_s}$. For this remainder selection, we apply the *FF* method, using the previously selected 1st order neighbors $\mathcal{S}^1_{a_c}$ as the preselected list $\mathcal{S}^{\text{pre}}_{a_c}$. For each selected neighbor, we add their sequence number and number of hops from a_s's position table $\mathcal{T}_{a_s}$.

When a beacon B_{a_s} is received, the current node a_c iterates through its position table $\mathcal{T}_{a_c}$, removing entries that were added via a_s and are no longer present in the beacon from a_s (B_{a_s}). Simultaneously, a_c also removes entries that have expired. This process ensures that $\mathcal{T}_{a_c}$ remains up to date by eliminating nodes from the table that are either missing from the recent beacon or have not been heard from within the validity interval. The validity interval $t_{validity}$, introduced in Section III-A, is defined as $t_{validity} = 2 \cdot \text{beacon interval}$. Here, $t_{validity}$ is set to be small to allow nodes to quickly respond to the loss of a link within their direct neighborhood. This approach maintains an accurate reflection of the latest network topology. In the subsequent step, for each neighbor a_i listed in B_{a_s}, the algorithm checks if a_i is already in $\mathcal{T}_{a_c}$. It updates a_i's information in $\mathcal{T}_{a_c}$ when the beacon provides a more recent sequence number or offers a shorter hop count with at least the same sequence freshness. If a_i is not found, it is added as a new entry to $\mathcal{T}_{a_c}$. For each new entry added via B_{a_s} in $\mathcal{T}_{a_c}$, a_s is set as the 'via node'.

D. Forwarding Mechanism

Let a_d denote the destination node. When a node a_c aims to send a packet to a_d, it is assumed that the position of a_d is known. For packet forwarding, a_c identifies a neighbor, a_{best}, within its position table $\mathcal{T}_{a_c}$ that offers the best advancement towards the destination. The packet is then directed to $a_j = \text{via}_{a_{\text{best}}}$, where a_j is a 1st order neighbor serving as the via address for a_{best}, ensuring maximal progress towards a_d. This approach is illustrated in Figure 4. To mitigate issues from delayed or lost beacons, a range reduction parameter ε is introduced. This parameter restricts selection to 1st order neighbors within a radius of $r - \varepsilon$, where $\varepsilon < r$ and r denotes the communication range. The selection of ε considers the worst-case scenario of two nodes at the edge of the communication range moving in opposite directions, defined as:

$$\varepsilon = 2 \cdot \text{beacon interval} \cdot \text{average speed}. \tag{2}$$

IV. Simulation Setup

This study evaluates greedy routing using subsets of the k-hop neighborhood with three selection methods denoted by Greedy-Random, Greedy-FF, and Greedy-EFF. Each scenario was repeated 50 times, using 95% confidence intervals.

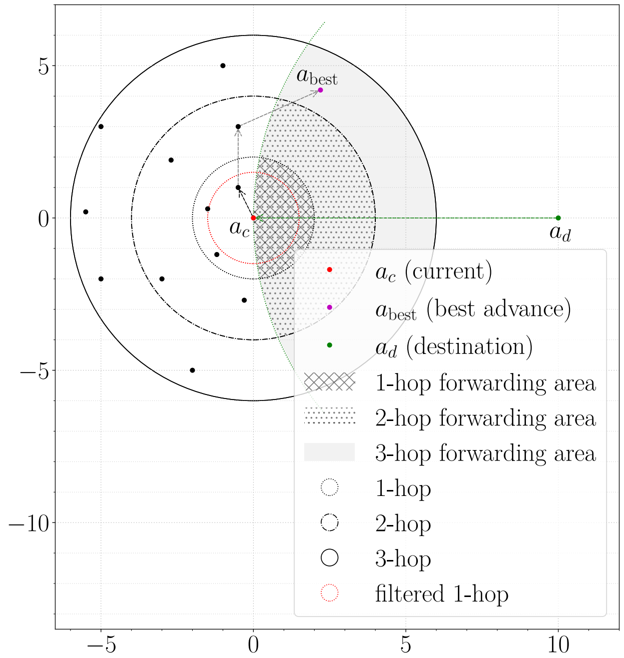


Fig. 4: Illustration of the packet forwarding process, where the packet is directed towards via$_{a_{\text{best}}}$, the node through which the best forwarder a_{best} is reachable.

A. Mobility Model

In this study, we model the French airspace, spanning 1 000 000 km^2, by approximating it with a 1250 km × 800 km rectangle as shown in Figure 5. Aircraft within this model can be at altitudes ranging from 0 km to 11 km, representing climbing, cruise, and descent flight phases. Based on aircraft density data from [11], the simulation incorporates on average 500 aircraft and includes one ground station, depicted in green in Figure 5. Each aircraft selects a random position within the simulation area and moves in a straight line in a randomly chosen direction. The LDACS communication range of 370.4 km is applied for A2G communication. We select an A2A communication range of 100 km, the same as evaluated in [11], which has been shown to achieve an approximate average ground station connectivity ratio of 100% in the French airspace, given its specific aircraft density.

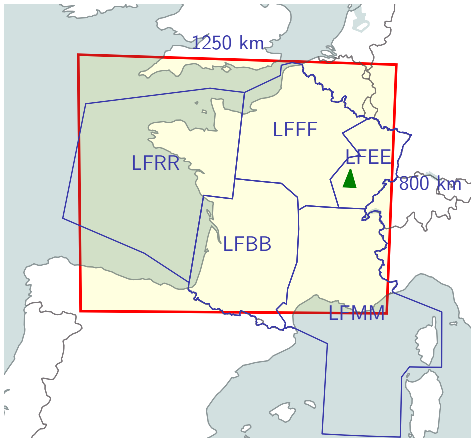


Fig. 5: The French airspace with its five Flight Information Regions (FIRs): Marseille (LFMM), Paris (LFFF), Brest (LFRR), Bordeaux (LFBB), and Reims (LFEE). The red outline represents the approximated rectangular boundary used in simulations. This figure is generated using [10].

B. Network Model

1) Data Application: As an application, we consider that aircraft employ Automatic Dependent Surveillance-Contract (ADS-C) technology to send their Four-dimensional (4D) positions (latitude, longitude, altitude, and time). This capability is essential for the safety and efficient management of future airspace [12]. Each aircraft sends an application packet of size 34 B every minute. The first transmission of each aircraft occurs randomly within a timeframe of [0 s, 60 s].

2) Data Link Layer: We implemented an abstract version of the MCSOTDMA, the proposed LDACS MAC layer, as specified in [4]. This implementation supports one LDACS transmitter and two receivers per aircraft, utilizing the SH channel for beacon broadcasting and the PP channels for unicast application packet transmission. The number of available PP channels dynamically adjusts based on location-specific blacklists, with the parameter set to 50 usable channels for this simulation [13]. Additionally, our abstract model results in a linear increase in MAC delay on the SH channel, proportional to the number of direct neighbors, mirroring MCSOTDMA behavior [6]. This model simplifies link establishment by assuming an entity with global node knowledge, which optimally schedules node transmissions on both SH and PP channels. Considering MAC headers as described in [4], a beacon on the LDACS A2A SH channel can accommodate up to 6 entries, each of size 12 B. Based on this maximum, we evaluated subset sizes $m = 4$ and $m = 6$ for the SH channel, with $m = 4$ identified as the optimal size in Section II.

3) Physical Layer: Our simulation employs a UDG radio model, assuming a uniform communication range across all aircraft and no channel errors. This model specifies that two nodes can successfully communicate only if they are within the defined communication range.

The simulation parameters are summarized in Table II.

V. Results and Discussion

In this experiment, we vary the equipage fraction, denoted by ρ, representing the proportion of air-

TABLE II: Simulation Parameters

Simulation Time Limit (s)	1800
Number of Simulation Runs	50, with 95% CIs
A2G Communication Range (km)	370.4
A2A Communication Range (km)	100
Number of Aircraft	500
Aircraft Speed (km/h)	800
Simulation Area (km^2)	1 000 000
Beacon Interval (s)	5
Slot Size (s)	0.024
Packet Size (B)	34
Application Sending Rate (pkts/node/min)	1
Aircraft Density ($\times 10^{-4}$ nodes/km^2)	2.5, 3, 3.5, 4, 4.5 and 5
LDACS Equipage Fraction ρ	0.5, 0.6, 0.7, 0.8, 0.9 and 1
Subset Size m	4 and 6

craft equipped with LDACS. We aim to evaluate the Greedy-Random, Greedy-FF, and Greedy-EFF algorithms at different equipage fractions. The performance of these algorithms is measured in terms of average PDR and hop count. We compare their performance against both Greedy-1 and Greedy-2. Although Greedy-2 includes the full neighborhood in the beacon, which is impractical as it exceeds the LDACS A2A slot capacity, it serves as a benchmark to assess the effectiveness of our proposed methods that utilize subsets of the neighborhood. Additionally, we also use Dijkstra's algorithm to find the shortest path, which assumes global knowledge. While the Dijkstra algorithm is not feasible here, we employ it to demonstrate the maximum achievable PDR and minimum hop counts for each given equipage fraction. Simulations are conducted using OMNeT++ with a communication range of 100 km and a range reduction parameter of $\varepsilon = 2.5$ km.

Greedy-Random, Greedy-FF, and Greedy-EFF improve the PDR by 1% compared to Greedy-1 at an equipage fraction of $\rho = 1$, which is the case when all aircraft are LDACS equipped, as illustrated in Figure 6. At $\rho = 0.5$, these improvements increase to 11.5% for $m = 4$ and 13% for $m = 6$. Additionally, Greedy-EFF marginally outperforms Greedy-FF on average in scenarios with lower equipage fractions due to its access to 3-hop neighborhood information. While Greedy-EFF shows a small advantage over Greedy-Random at $m = 4$, increasing to $m = 6$ provides minimal additional gains. Overall, all methods with subset sizes $m = 4$ and $m = 6$ achieve the performance gains comparable to Greedy-2. Despite achieving higher PDR over Greedy-1, a performance gap in PDR of at least 10% compared to Dijkstra persists for all methods at an equipage fraction of $\rho = 0.5$. This discrepancy arises from the lack of a fallback mechanism to navigate through dead-end situations, leading to packet drops.

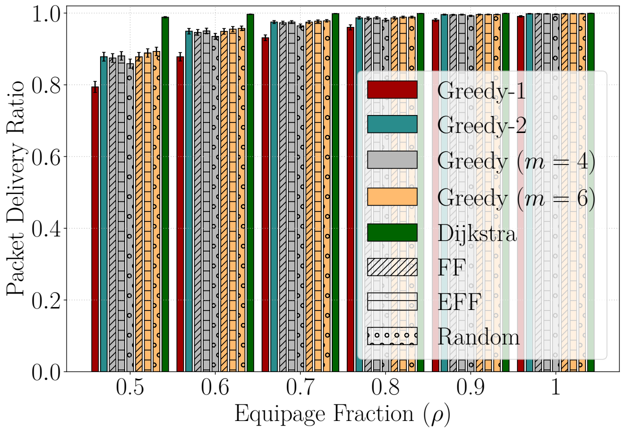


Fig. 6: Average PDR across different methods as a function of equipage fraction.

The average hop count results over different equipage fractions are shown in Figure 7. At an equipage fraction of $\rho = 1$, Greedy-Random shows slightly higher hop counts than its counterparts because it selects neighbors without considering their positions. As ρ decreases, this discrepancy diminishes. Overall, the hop counts for all our proposed methods are closely aligned with those of Greedy-2. Additionally, all greedy approaches align with the Dijkstra shortest path algorithm and have hop counts that are higher by up to 0.55 hops on average for an equipage fraction of $\rho = 1$.

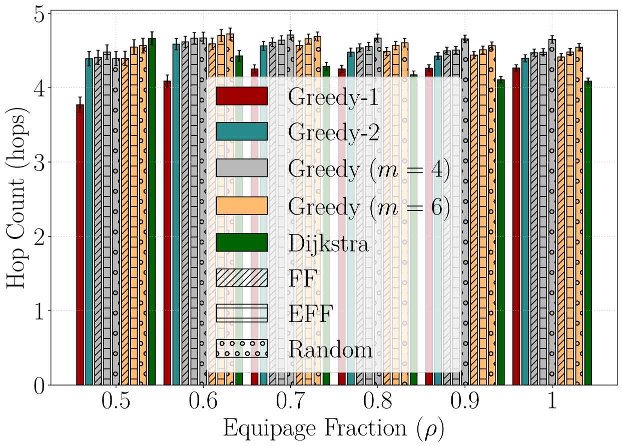


Fig. 7: Average hop count across different methods as a function of equipage fraction.

Following the previous analysis where Greedy-Random, Greedy-FF, and Greedy-EFF showed roughly similar average PDR and hop counts, we now evaluate the *Random*, *FF* and *EFF* algorithms in terms of selecting k^{th} order neighbors. We define the Capture Ratio as the ratio of unique k^{th} order neighbors identified by each method against the actual total, averaging this across all nodes. Our simulations involve 50 random mobility snapshots conducted in an area large enough to mitigate edge effects, reflecting the density and equipage fractions from the previous scenario.

As outlined in Section II and shown in Figure 6, employing a subset size of $m = 4$ achieves a balance of coverage gain over overlap. We now present how this subset size influences the average capture ratio of 2nd and 3rd order neighbors. The results in Figure 8 indicate that at a node density of 5×10^{-4} nodes/km^2 and with $m = 4$, the *FF* method achieves a capture ratio of 0.6 for 2nd order neighbors, compared to 0.52 by Random, marking a 15.4% improvement.

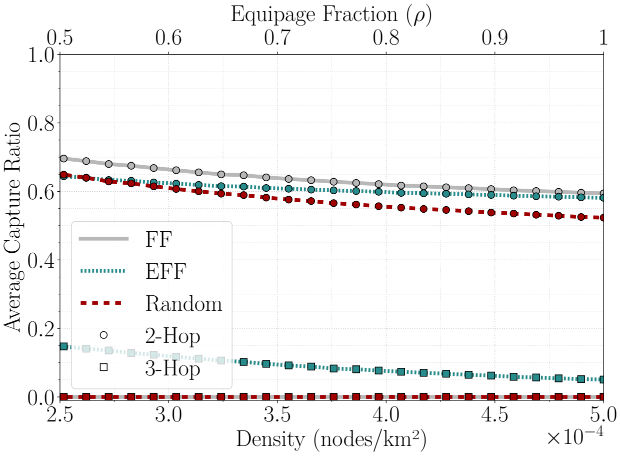


Fig. 8: Average k^{th} order neighbors capture ratio for 100 km communication range with subset size $m = 4$.

Random selection becomes less effective at higher densities due to an enlarged selection pool. At a density of 2.5×10^{-4} nodes/km^2, the *FF* method achieves a capture ratio of 0.7, compared to 0.64 by Random, reflecting a 9.3% improvement. With only 7.6 first-order neighbors on average, selecting just four means Random's performance remains effective. Additionally, at this density, *EFF* achieves the same performance as *Random* by strategically sacrificing some 2nd order neighbors to capture 15% of 3rd order neighbors, a capability not present in *FF* or *Random*. At a higher density of 5×10^{-4} nodes/km^2, despite selecting fewer 1st order neighbors than *Random*, *EFF* achieves a 2nd order neighbor capture ratio of 0.582, compared to 0.52 by Random, marking a 12% improvement, and it also captures 5.1% of 3rd order neighbors. As density increases, *EFF*'s performance is expected to align more closely with *FF* because the distance threshold used in *EFF* will have minimal or no effect.

With $m = 6$, the average 2nd order neighbors capture ratios for *FF*, *EFF*, and *Random* range between $[0.84, 0.7]$, $[0.70, 0.61]$, and $[0.83, 0.67]$ respectively, as demonstrated in Figure 9. Additionally, *EFF* captured 34.6% of 3rd order neighbors at a density of 2.5×10^{-4} nodes/km^2 and 25.8% at 5×10^{-4} nodes/km^2.

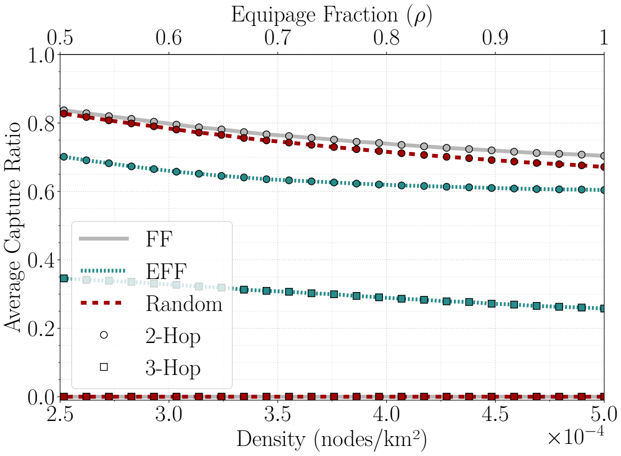


Fig. 9: Average k^{th} order neighbors capture ratio for 100 km communication range with subset size $m = 6$.

VI. Related Work

Sparse networks exacerbate dead-end situations, leading to the exploration of greedy forwarding strategies that include 2-hop or 3-hop neighborhood information [9], [14]. Moreover, it is found that the performance gain from using Greedy-2 is higher than that achieved by further expanding to Greedy-3 [14]. While this approach mitigates routing inefficiencies, it significantly increases beacon size, an issue not fully addressed in previous studies. The slot-based MAC protocol in LDACS A2A communications further complicates beacon size management within strict slot durations. Studies like [15] and [8] show including direct neighbors in the beacon doubles the overhead ratio compared to using only Greedy-1, without considering beacon size reduction necessary for slot duration compliance.

Addressing the beacon size challenge, existing literature aimed to reduce message counts for capturing 2-hop neighborhood information using $O(n)$ messages of $O(\log n)$ bits each, where n is the number of nodes in the network [16]. This approach focuses on message efficiency rather than direct beacon size reduction. Further advancements utilized probabilistic structures like Bloom filters to compress the neighbor list [17]–[19]. These methods efficiently encapsulate all direct neighbors and, by extension, can accommodate k-hop neighborhood. Nonetheless, the primary motivation behind these works was to enhance broadcasting protocols, not to support geographic Greedy-k routing, thus lacking the positional information required for it.

Our investigation proposes a solution addressing both the need for a fixed maximum beacon size including a subset of direct neighbors with positional information and the constraints of LDACS A2A MAC, which accommodates approximately an additional 72 B in the beacon after headers. This constraint influences the feasible number of neighbors in a beacon, based on entry size, e.g., 12 B in our case as specified in Section III-A. Our method aims to select nodes within this subset to optimally enhance performance, achieving Greedy-2 forwarding gains within LDACS A2A communication slot durations. To the best of our knowledge, our approach is the first to reduce beacon size by selecting a subset of the neighborhood and including both MAC addresses and positional information in the beacon, offering a novel contribution to the field.

VII. Conclusion

We presented a novel approach to enhance greedy forwarding in sparse networks by leveraging 3-hop neighborhood information. We propose three methods for selecting subsets of k^{th} order neighbors: *Random*, *FF*, and *EFF*. Additionally, we analytically derive an optimal fixed subset size that achieves performance improvements with Greedy-2 forwarding. We find that a subset size of 4 is optimal, providing performance gain while maintaining the rate of change of coverage higher than the rate of change of overlap.

We performed a simulation study to evaluate the performance of our proposed routing protocol and the selection methods over the French airspace with different equipage fractions. We investigated the average PDR, hop count, and capture ratio of k^{th} order neighbors. Our method proves superior to Greedy-1, especially in improving the packet delivery ratio in sparse scenarios where dead-ends occur more frequently. Our evaluations demonstrate that all the proposed methods, on average, enhance the PDR by up to 13% relative to Greedy-1, providing the performance gain of Greedy-2. Furthermore, Greedy-FF captures more 2^{nd} order neighbors on average compared to Greedy-Random and Greedy-EFF. Additionally, Greedy-EFF can capture up to 34.6% of 3^{rd} order neighbors on average, while the other methods primarily focus on 2^{nd} order neighbors. All models were implemented using the OMNeT++ simulation tool and Python, and details of this implementation along with the simulation model are available as an open-source for the research community in [20].

This approach could provide potential benefits for geographical routing in applications beyond aeronautical communication, particularly in similar low-density large networks where greedy forwarding more frequently reaches dead-ends. Enhanced neighborhood awareness could be applied in various applications such as load balancing by capturing the load status of 2^{nd} and 3^{rd} order neighbors. Furthermore, our method's partial awareness of the k-hop neighborhood lays the groundwork for developing effective strategies when a k-hop neighborhood is insufficient. This method can be generalized for k-hop neighborhoods and 3D networks by adapting the subset size m.

References

[1] Eurocontrol, "Aviation Outlook 2050," Main Report, Apr. 2022.

[2] "White paper: Ubiquitous aviation connectivity with LDACS," FREQUENTIS AG, Technical Report, 2021.

[3] M. A. Bellido-Manganell and M. Schnell, "Towards Modern Air-to-Air Communications: The LDACS A2A Mode," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, Sep. 2019, pp. 1–10.

[4] S. Lindner, K. Fuger, M. A. E. Ahmed, A. Timm-Giel, J. Hampel, and M. Bellido, "MCSOTDMA Protocol Specification," Zenodo, Tech. Rep., Jun. 2023.

[5] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A Geographic Routing Strategy for North Atlantic In-Flight Internet Access Via Airborne Mesh Networking," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1231–1244, Aug. 2012.

[6] S. Lindner, K. Fuger, M. A. E. Ahmed, and A. Timm-Giel, "Multi-Channel Self-Organized TDMA for Future Aeronautical Mobile Ad-Hoc Networks," *IEEE Transactions on Vehicular Technology*, pp. 1–15, 2024.

[7] D. Medina, F. Hoffmann, F. Rossetto, and C.-H. Rokitansky, "A Geographic Routing Strategy for North Atlantic In-Flight Internet Access Via Airborne Mesh Networking," *IEEE/ACM Transactions on Networking*, vol. 20, no. 4, pp. 1231–1244, Aug. 2012.

[8] M. Y. Arafat and S. Moh, "A Q-Learning-Based Topology-Aware Routing Protocol for Flying Ad Hoc Networks," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1985–2000, Feb. 2022.

[9] J. Zhou, Y. Chen, B. Leong, and P. Sundaramoorthy, "Practical 3D geographic routing for wireless sensor networks," in *SenSys 2010 - Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 2010, pp. 337–350.

[10] X. Olive, "Traffic, a toolbox for processing and analysing air traffic data," *Journal of Open Source Software*, vol. 4, no. 39, p. 1518, Jul. 2019.

[11] Q. Vey, A. Pirovano, J. Radzik, and F. Garcia, "Aeronautical Ad Hoc Network for Civil Aviation," in *Communication Technologies for Vehicles*, A. Sikora, M. Berbineau, A. Vinel, M. Jonsson, A. Pirovano, and M. Aguado, Eds., Cham: Springer International Publishing, 2014, pp. 81–93.

[12] J. Valle Martinez, "Network 4D Trajectory CONOPS," Technical Report 23/08/28/46, Sep. 2023.

[13] M. A. Bellido-Manganell and M. Schnell, “Feasibility of the Frequency Planning for LDACS Air-to-Air Communications in the L-Band,” in *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Apr. 2021, pp. 1–14.

[14] C. S. Chen, Y. Li, and Y.-Q. Song, “An exploration of geographic routing with k-hop based searching in wireless sensor networks,” in *2008 Third International Conference on Communications and Networking in China*, Aug. 2008, pp. 376–381.

[15] Y. Li, C. S. Chen, Y.-Q. Song, Z. Wang, and Y. Sun, “Enhancing Real-Time Delivery in Wireless Sensor Networks With Two-Hop Information,” *IEEE Transactions on Industrial Informatics*, vol. 5, no. 2, pp. 113–122, May 2009.

[16] G. Calinescu, “Computing 2-Hop Neighborhoods in Ad Hoc Wireless Networks,” in *Ad-Hoc, Mobile, and Wireless Networks*, S. Pierre, M. Barbeau, and E. Kranakis, Eds., ser. Lecture Notes in Computer Science, Montreal, Canada: Springer, 2003, pp. 175–186.

[17] K. C. Lee, U. Lee, and M. Gerla, “Geo-opportunistic routing for vehicular networks,” *IEEE Communications Magazine*, vol. 48, no. 5, pp. 164–170, May 2010.

[18] K. Na Nakorn, Y. Ji, and K. Rojviboonchai, “Bloom Filter for Fixed-Size Beacon in VANET,” in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, Seoul, South Korea: IEEE, May 2014, pp. 1–5.

[19] F. Klingler, R. Cohen, C. Sommer, and F. Dressler, “Bloom Hopping: Bloom Filter Based 2-Hop Neighbor Management in VANETs,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 534–545, Mar. 2019.

[20] *LDACS Greedy K-Hop Simulator (removed for reviewing process.)*